International Journal of Electronics,
Communication & Instrumentation Engineering
Research and Development (IJECIERD)
ISSN(P): 2249-684X; ISSN(E): 2249-7951
Vol. 4, Issue 3, Jun 2014, 143-150

EVALUATION OF EFFECT OF BUFFER SIZE ON NOC PERFORMANCE

MOSTAFA HAGHI[1], M. ASHA RANI[2] & ELHAM JAVADI[3]

[1,2]Department of Electronics and Communication, JNT University, Hyderabad, Andhra Pradesh, India

[3]Department of Electronics and Computer Engineering, Tehran University, Tehran, Iran

ABSTRACT

When designing a System-on-Chip (SOC) using a Network-on-Chip (NOC), delay and throughput are two key elements to optimize. Packet Injection Rate (PIR) and buffer size are two vital parameters that have direct effect on the performance of the NOC. Buffer size, play a very important role in order to decrease the number of delay cycles and improve the Throughput. In this paper we study the effect of buffer size increasing on NOC with FULLY ADAPTIVE routing algorithm. In this case study the mentioned algorithm has been chosen due to its sensitivity to PIR. The area of research is partitioned in 3 sections, It is observed that while the PIR is limited in 0.01<PIR<0.03 range, with increasing the size of buffer the time latency is decreased and Throughput is improved significantly, for the 0.03<PIR<0.05 the positive effect of the Buffer size is reduced and when the PIR cross the 0.05, is seen that with increasing the size of buffer behavior of NOC regard to Delay and Throughput is not predictable.

Mesh topology is applied with 16 cores (N*N), and performance evaluation is based on flit-accurate and open source system C simulator.

KEYWORDS: PIR, Buffer Size, Fully Adaptive, NOC, SOC

INTRODUCTION

System On Chip(SOC) in upcoming billion transistors era for two major reasons of power density limitation and technology improvement involve the integration of numerous heterogeneous semiconductor source blocks[1][2]. This source block can be any of: processor, FPGA, Memory or any other intellectual property (IP).

Networks on Chip [3] (NOCs) have been taken in consideration lately specially in the last few years as a promising alternative to bus-based and point-to-point architectures to interconnect Intellectual Property cores (IPs) in Systems on Chip (SOCs). NOCs allow transactions between several pairs of cores and the connection of new IPs does not imply redesigning the communication infrastructure at the same time and cause no significant performance reduction on the overall system, all these points make the NOC paradigm popular and conventional in lately SOCs design.

A routing algorithm is defined as a path taken between any two sources and destination nodes by a packet, according to where the decision about routing path is taken the routing is classified into category of a) Source b) distributed [4].

In source the entire path is known to packet before leaving the first node, while in distributed routing each node receives the packet from previous one and over there regards to network state make a decision where to be the next IP destination. According how a direction is defined to send a packet to next IP routing is classified into: a) Deterministic b) Adaptive [5].

Fully Adaptive Algorithm

The fully adaptive routing algorithm always uses the route that is not congested; this is a policy in this routing algorithm. While the distance in other type of routing algorithms is a crucial point, in this routing is not a matter, it means that the routing algorithm choose a direction between sender and receiver IPs, although it is not the shortest way. Typically an adaptive routing algorithm sets alternative congestion free routes in order to superiority, of course shortest direction is the best one [6].

Buffer Size

Network-on-Chip (NOC) architectures are becoming an important fact to fabrication for both general-purpose chip multi-processors (CMPs) and application-specific systems-on-chips (SOCs). In the design of NOCs, high throughput and low time latency are both important design parameters that have to be taken into consideration; the router micro architecture plays a vital role in achieving these performance goals. High throughput routers allow an NOC to satisfy the communication requirements of multi cores and many cores applications, or here a trade off can be done among higher throughput, power and bandwidth, it is achieved in this way that the higher throughput is achieved with power saving while fewer resources are being used to attain a target bandwidth. Router micro architecture plays a central role in the performance of an on-chip network (NOC) [7]. In an NOC buffers are required in routers to host incoming flits which cannot be immediately forwarded to next IP due to congested. The buffering may be done at the inputs or the outputs of a router, corresponding to which one is occurred it is called an input-buffered router (IBR) or an output-buffered router (OBR). OBRs are more attractive because they can sustain higher throughputs and have lower queuing delays under high loads than IBRs. The MIN size of a buffer has to be equal or greater than a packet size.

PIR

The rate that packets are injected into network by an IP is named Packet Injection Rate (PIR), unit for PIR is known with Packets/cycle/IP. PIR can have crucial role in improving the performance of the network. It is restricted between 0 and 1. For example when PIR=0.05, it means that each node sends 2 packets every 10 cycles [8].

RELATED WORKS

This Section discusses about works regard to buffer sizing and its relation with QOS parameters. Different methods have been proposed for buffer management and buffer management.Hu et al. [9] presented a method to manage size of buffer on intermediate nodes of NOCs, using queuing theory formalisms. The main point is to minimize the average latency of all communications that occur on the NOC, by reducing the area that is occupied by buffer. The Authors considered data storage using packets as atomic unit, i.e. store-and-forward switching mode. The store-and-forward technique is not frequently used in NOCs, since it increases Delay and area, because NOC buffers size must be at least the size of the maximum size packet.

In terms of traffic modeling, the Authors employ a Poisson synthetic model to characterize communication networks. The disadvantage of this model is that, it reduced accuracy compared to trace-based models or even self-similar models. Varatkar and Marculescu [10] demonstrated the self-similar characteristic of MPEG traffic, one typical application on current SOCs. The Authors proved that it is possible to define the optimal size for buffers of MPEG decoder modules to avoid buffer overflow.

In terms of traffic modeling, a method for synthetic traffic generation is presented where traces of traffic and their statistical properties are combined on a synthetic trace generation procedure. However, experimental traffic scenarios considered only point-to-point communication, discarding the possible influence of concurrent flows. In addition, the Authors give no data about the buffer threshold value. Chandra et al. [11] presented a method to size buffers considering data production and consumption rates of packets transmitted in burst.

Throughput is the performance metric employed to compare atomic and distributed buffers. Atomic buffers are those at target IPs, while distributed buffers are those found on intermediate nodes. Results presented in [11] show higher throughput for the distributed buffer strategy. The main weakness of this method is that the NOC is optimized for a fixed traffic scenario, which is inadequate to SOCs that accept applications defined after design.

EXPERIMENTAL RESULTS

In this paper the work is started with PIR=0.01 and Buffer size=1, in first step PIR is increased and the Buffer size is kept fixed, then delay and Throughput results are recorded, in second step we apply the worst case of PIR in each period to simulator and size of buffer is extended and the results are recorded for delay and throughput, therefore by comparing the results in both situations we can conclude that by extension of buffer size how much performance is improved.

a. 0.01<=PIR<=0.027, Buffer size=1

Simulation is begun with mentioned amount PIR and B.S above; PIR's step is incremented by 0.01. It's observed that according to Figure 1 as the PIR is incremented, the two parameters of Delay and Throughput also are raised. The peak of Throughput is occurred at PIR=0.02 and then for PIR>0.02 the both studied parameters are getting worse. The MAX amount of PIR for Buffer size=1 will occur at PIR=0.027, this size of buffer will not support PIR>0.027 any longer and NOC is dumped. It's remarkable that the worst Delay case is happened in this PIR. As was calculated before Delay Cycles greater than 26 are not acceptable here.

The worst state of PIR while B.S=1 is equal to 0.027, this amount is taken into consideration, in next step the B.S is extended to 2(B.S=2). While the B.S is increased 100% in size, Throughput is improved by 300.33% and Delay is reduced by 67.1%, this is very significantly improvement, these results are expected because rate of packet injection into IPs are increased and on the other hand the directions betweens IPs are not congested yet due to buffer size extension. We can continue extension of B.S, but it seen that the slope of Delay figure in Figure 2 is reduced but still Delay cycles are decreasing and Throughput is improved the reason of alleviation of Delay figure is that, ratio of B.S to PIR with increasing of PIR is getting down while B.S is fixed, consequently regards to applied algorithm, the NOC is going to be contented, with increasing the B.S=8 saturation state occurs, while B.S has been extended for 700% in size, Delay cycles are reduced from 31.62 to 18.52 (70.7%) and the Throughput is improved from0.6 to 16.2, it means 27 times of worst Throughput case (2600%), this is a promising improvement, but we have to remember the rate of Delay and Throughput improvement is reduced with adding steps to PIR.

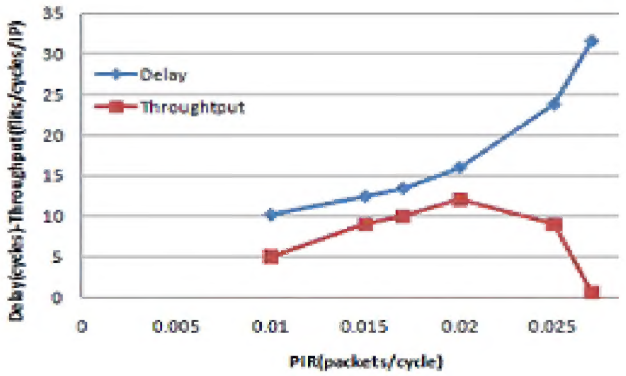

Figure 1: Delay and Throughput under PIR Variant (B.S=1)

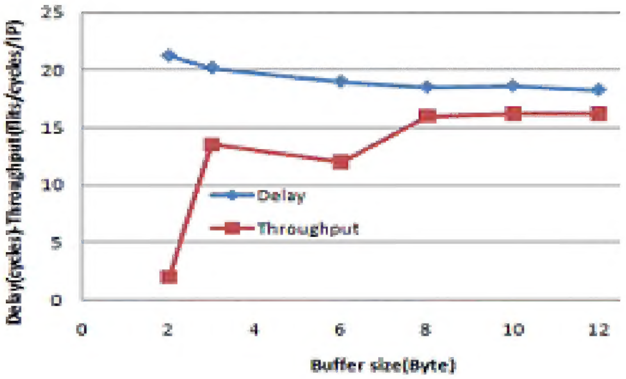

Figure 2: Delay and Throughput under B.S Variant (PIR=0.027)

b. 0.028<=PIR<=0.030, B.S=4

At the first place it is important to say that the MIN amount of B.S for PIR=0.028 to support the network is 2, but it's not trustable, it is very likely to fail with these parameters: PIR=0.028 and B.S=2.

Therefore for 0.028<=PIR<=0.030 we focus on B.S=4, at this B.S and at PIR=0.028 the Delay and Throughput are 21.56 and 14 respectively, It is possible to increase the PIR to 0.030,its still supported by B.S=4 but no improvement in Throughput nor in Time latency as is drawn in Figure 3 (Delay=28.1 & Throughput=0.6), the reason is the same with previous case, means ration of PIR to B.S is reduced and communication between modules becomes tough but still ongoing in cost of Time latency.

We record the PIR=0.030 as the worst case for B.S=4, now we extend the B.S from 4 to 6 means 50% to see what will occur, by this alteration its observed that the Throughput is increased 30 times (2900%) and Delay 22.9%. Buffer size can be extended more but it is almost in saturation state for B.S>6, if B.S is incremented from 6 to 16(260%) gradually, Delay is reduced only 3.9%and Throughput just -0.5% (Figure 4). It is concluded that the best Buffer Size for 0.028<=PIR=<0.030 is 6. For B.S>6, PIR is not sensitive to B.S extension.

This early saturation refers to limited number of IPs in this NOC. Has to be reminded, applied routing algorithm is Fully adaptive routing algorithm always looking for not congested directions, where here the number of modules are restricted to 16, with increasing the PIR network faces with congested direction quickly that even by extending the B.S can't overcome on it.

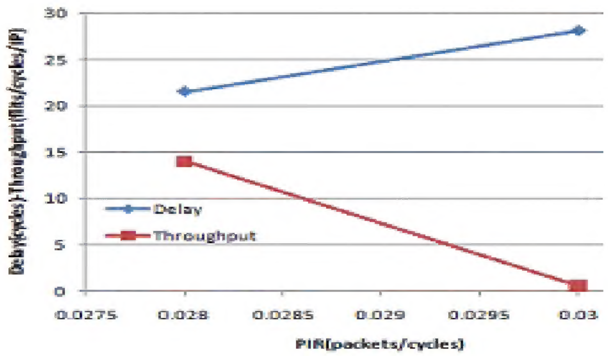

Figure 3: Delay and Throughput under PIR Variant (B.S=4)

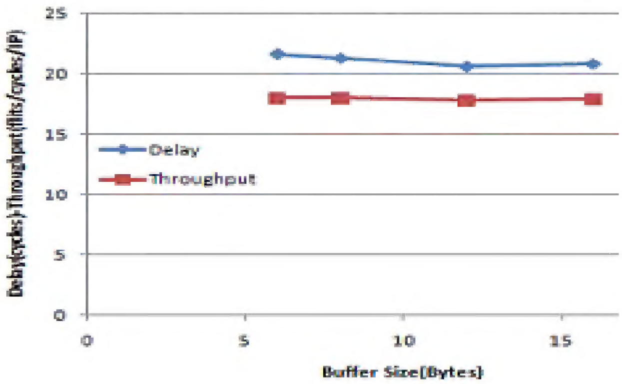

Figure 4: Delay and Throughput under B.S Variant (PIR=0.030)

c. 0.031<=PIR<=0.034, B.S=6

The routine policy is when PIR is increased size of buffer has to be extended otherwise simulation is failed. The MIN amount of B.S for PIR=0.030 is 6 to simulator runs. When B.S=6 and PIR=0.030 Delay and Throughput are 23.31 and 18 respectively, by increasing PIR to 0.033 and 0.034 Time latency will be 27.15 and 29.64 respectively while Throughput is equal to 2.6 for PIR=0.033 and 1.2 for PIR=0.034(Figure 5).It is observed that the Throughput is reduced dramatically as the PIR is increasing, even extending the size of buffer cannot alleviate it effectively, in this status NOC is in saturated state, In second step we focus on worst PIR of first level(PIR=0.034) and extend the B.S from 6 to 9 (50% in size). The result is decreasing the Time latency as much as 15% (from 29.64 to 25.11). B.S is extended two more times, in first time 80% and in second attempt 100%, consequently it is seen that Delay is reduced to 25.1 and 24.8 and very significant improvement is occurred in Throughput, in B.S=10, 12 Throughput is 20(same for both) it means that the directions among IPs have been congested and saturation state has happened consequently (Figure 6).

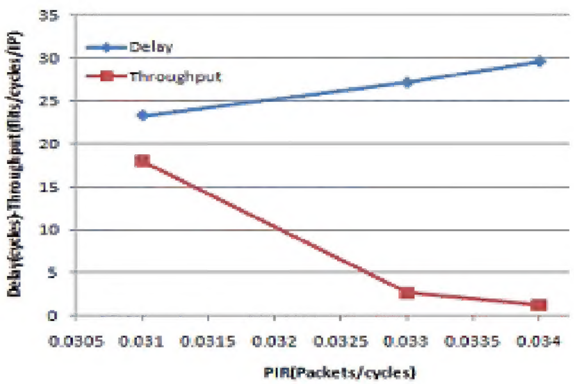

Figure 5: Delay and Throughput under PIR Variant (B.S=6)

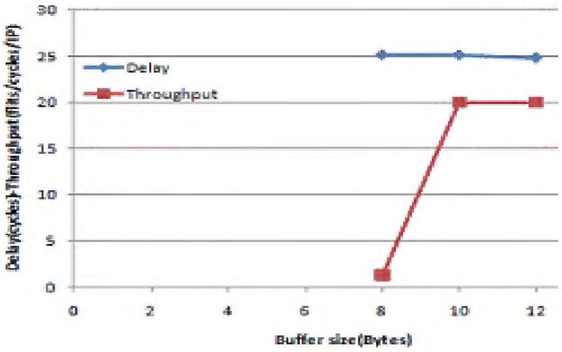

Figure 6: Delay and Throughput under B.S Variant (PIR=0.034)

d. 0.035<=PIR<=0.037, B.S=8,

The policy is same, two steps, varying the PIR and record Delay and Throughput and then consider about the worst PIR amount and extending the B.S then comparing the improvement. We can observe that Time latency is getting higher and Throughput is reducing gradually while PIR is incremented this has been the routine behavior of the NOC. In this case also results for PIR=0.035, 0.036, 0.037 are as follow Delay=27.9, 30.1, 35.3 and Throughput=2, 8, 1. The worst PIR is 0.037, now it's time to extend the B.S to 12(50%).

The results are raised up for B.S=12, 16, 20

Delay = 30, 30.64, 28

Throughput = 22, 22.2, 22.2

Throughput is almost fixed and Delay also is reduced slightly. Encountering with these results and analyzing them, force us to conclude that extending the B.S is not useful as much as it was at lower rate of PIR any longer and the capacity of network's channels is fully contented.

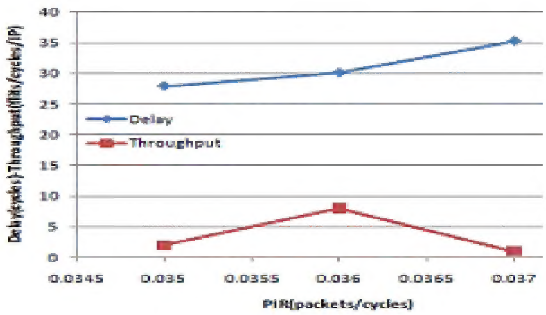

Figure 7 Delay and Throughput under PIR Variant (B.S=8)

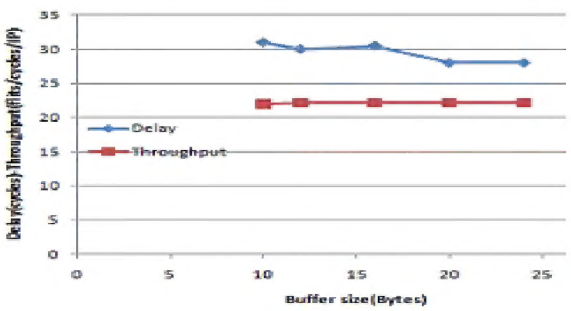

Figure 8: Delay and Throughput under B.S Variant (PIR=0.037)

e.PIR>0.037….B.S=10, 12, 16, 24, 48, 60, 72

In table 1 we observe that as the PIR is increasing, the rate of Delay is getting high with a sharp slope and its out of control and unacceptable, on the other hand Throughput is fall down for PRI>0.051. The remarkable point is that if we want to still keep a balance between B.S and PIR, the size of buffer has to be extended with a sharper rate than PIR rate.

In table 2 we have tried to extend the Size of buffer to alleviate the negative effect of high PIR, but still it is not effective. In table.3 obviously is seen that when PIR crosses the 0.055 there is no order in Delay and Throughput and even in some cases it is seen that with extending the B.S the Delay is reduced slightly.

Table 1

PIR	0.038	**0.039**	**0.040**	0.041	0.042	0.043	0.045	0.047	0.049	0.050	0.051	0.052	0.054	0.055	0.058
Delay	31.97	34.1	37	37.1	39.9	42.2	52.1	57.6	115.7	91.5	102.5	273	333	488.2	824.2
Throughput	22.7	23	24	24	25	24	26	3.2	1.3	29	30	1	1	0.7	0.2
B.S	12	12	16	24	24	24	24	24	24	72	72	72	72	144	144

Table 2

PIR	0.039	0.039	0.039	0.049	0.049	0.049	0.054	0.054	0.054
Delay	32.64	32.2	32.7	103.31	77.6	84	310.6	250.3	336
Throughput	23.2	23.3	23.3	8	19	29	2	1.3	3.6
B.S	18	24	30	30	36	60	108	120	132

Table 3

PIR	0.058	0.058	0.058	0.058	0.058	0.058
Delay	695.76	1110	650	698.9	703.3	940
Throughput	0.1	3	0.8	1.1	2.2	5
B.S	216	288	350	550	570	630

Comparing

In following in Figure 9, relation among Delay, Throughput and B.S is investigated. If we consider in period of 0.01<PIR<0.040 we realize that the slope of three figures are almost the same, it means that while the PIR is increased in network, by extension of B.S with same rate in contrast with PIR we can control the Time latency and Throughput, but for PIR>0.040 but less than 0.050 if we are going to control the Throughput in acceptable range, the rate of B.S extension has to be increased but there is no hope to keep Delay in expected level. When the PIR crosses 0.050 the situation gets even worse and all parameters are out of control, in fact Delay is increased with following no pattern and Throughput also falls down, in this situation extension of B.S doesn't work.

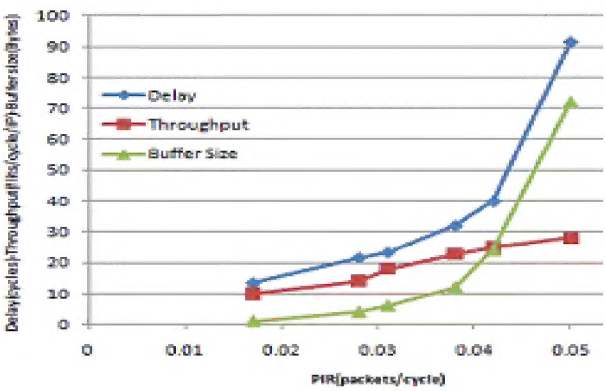

Figure 9: Comparing of Delay, Throughput and B.S under PIR Variation

CONCLUSIONS

In this paper behavior of a 16(4*4) core NOC under PIR variation and B.S extension has been studied. In fact the range of PIR is varying between 0.01and 0.058 and B.S also is variable from 1 to 288, since the number of cores in this study are limited, by increasing the PIR, buffer will be congested, consequently Delay is increased and Throughput is reduced, to alleviate this matter we need to extend size of buffer. This technique works well while PIR is varying between 0.01 and 0.034, but for PIR between 0.035and 0.054 the B.S has to be extended with higher rate and has lower effect, eventually for PIR>0.054 even with extension of B.S it is impossible to control the NOC and core will be dumped. As in this work the studied Network is restricted to 16, by increasing the PIR rate, congested direction is occurred soon, In this situation we suggest that for 16 cores network on chip with Fully Adaptive routing algorithm the PIR has to be kept under 0.034 for the best performance.

REFERENCES

1. L. Benini and G. De Micheli, “Networks on chip: a new paradigm for systems on chip design,” Proc. of Design, Automation and Test Conference in Europe, pp. 418-419, 2002.
2. W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” Proc. of Design Automation Conference, pp. 684-689, 2001.
3. L. Benini and G.D. Micheli, “Networks on Chips: A New SoC Paradigm,” Computer, vol. 35, no. 1, pp.70-78, Jan. 2002.
4. Benini, L.; De Micheli, G. Networks on chips: a new SoC paradigm. IEEE Computer, v. 35(1), Jan. 2002, pp. 70-78.
5. Yuan-Long Jeang, Tzuu-shaang Wey, Hung-Yu Wang, Chung-Wei Hung, and Ji-Hong Liu, 2000. An Adaptive Routing Algorithm for Mesh-Tree Architecture in Network-on-Chip Designs.
6. G. M. Chiu, “The odd-even turn model for adaptive routing,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, 2000, pp. 729-738.
7. J. Hu and R. Marculescu, “Application specific buffer space allocation for networkson- chip router design,” in *Proceedings of International Conference on Computer Aided Design*, 2004, pp. 354-361.
8. T. Marescaux, A. Rangevall, V. Nollet, A. Bartic, and H. Corporaal, “Distributed congestion control for packet switched networks-on-chip,” in *Proceedings of Parallel Computing Conference*, 2005, pp. 760-768
9. Hu, J.; Ogras, U. Y.; Marculescu, R.. “System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design”. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25-12, December 2006, pp. 2919-2933.
10. Varatkar, G.; Marculescu, R. “On-Chip Traffic Modeling and Synthesis for MPEG-2 Video Applications”. IEEE Transactions on VLSI Systems, 12-1, January 2004, pp. 108-119.
11. Chandra, V.; Xu, A.; Schmit, H.; Pileggi, L. “An interconnect channel design methodology for high performance integrated circuits” in Proceedings of the Design, Automation and Test in Europe, DATE’04, 2004, pp. 1138-1143.